Peer-to-peer protocol facilitates real-time communications

The low communications overhead of the CSMA/DCR peer-to-peer protocol makes it desirable for real-time network applications. Further, the protocol's guaranteed network-access time relieves the designer from worst-case haunts.

Deif N Atallah, *Intel Corp*

By definition, real-time-communication networks need to transfer information between nodes as quickly as possible. Preferably, these transfers should occur within a guaranteed maximum time limit. Networks have traditionally achieved real-time communications via some type of master/slave protocol, contention-based protocol, or token-based protocol—each with their own advantages and disadvantages. A relatively new contention-based protocol, based on the advantages of each of these methods, enhances the information-transfer rate of a real-time network.

A master/slave protocol is a popular method for real-time communications because it guarantees that all the nodes on the network will gain access. Examples include the Synchronous Data Link Control (SDLC) and High Level Data Link Control (HDLC) protocols. A master/slave protocol requires the management of a significant amount of overhead, however, which in itself occupies valuable bandwidth. A master node has to first poll the slave nodes to initiate communication, and it must then take control of the link to ensure that communication takes place. If the network has multiple masters, they must arbitrate for control of the network, so that a maximum access time cannot be guaranteed.

Token-passing protocols, in contrast, guarantee a maximum access time for each node on the network by sequentially passing a token around to each node. To avoid contention, each node must acquire the token before transmitting information. Even if the network is quiet, however, a node must wait until the token goes all the way around the network. If there are many nodes on the network, the guaranteed maximum access time can be long. Examples of token-passing protocols are the IEEE 802.4 and 802.5.

A peer-to-peer communication protocol such as the Carrier Sense Multiple Access with Collision Detection (CSMA/CD), which is what Ethernet uses, has low communications overhead. It can't guarantee a maximum access time to the network, however, because it resolves contention problems statistically. The CSMA/CD lets a node monitor the network for a quiet period in which to send its message. If messages from two nodes collide, the two nodes have to wait for a random interval of time and then they attempt to retransmit. Although you can choose network parameters to minimize the

The CSMA/DCR protocol guarantess a maximum access time to the network, yet it has the same low communications overhead as the CSMA/CD.

probability of successive collisions, you can't be guaranteed of a maximum access time because of the protocol's nondeterministic recovery mechanism.

Determinism offers a guarantee

A variation of the CSMA/CD protocol, the CSMA/DCR (deterministic collision resolution), merges the advantages of the aforementioned protocols. The protocol features the same low communications overhead as the CSMA/CD, yet guarantees a maximum access time to the network.

The protocol operates in a peer-to-peer CSMA/CD mode until a message collision occurs. Any node on the network can initiate communications without polling for a response. This lack of polling overhead maximizes the link's bandwidth. Essentially, the physical layer of the OSI (open system interconnection) model looks for opportunities to transmit by monitoring the network's activity. This network-access scheme, coupled with the guaranteed access time of the resolution phase, produces a very efficient network even during heavy traffic conditions.

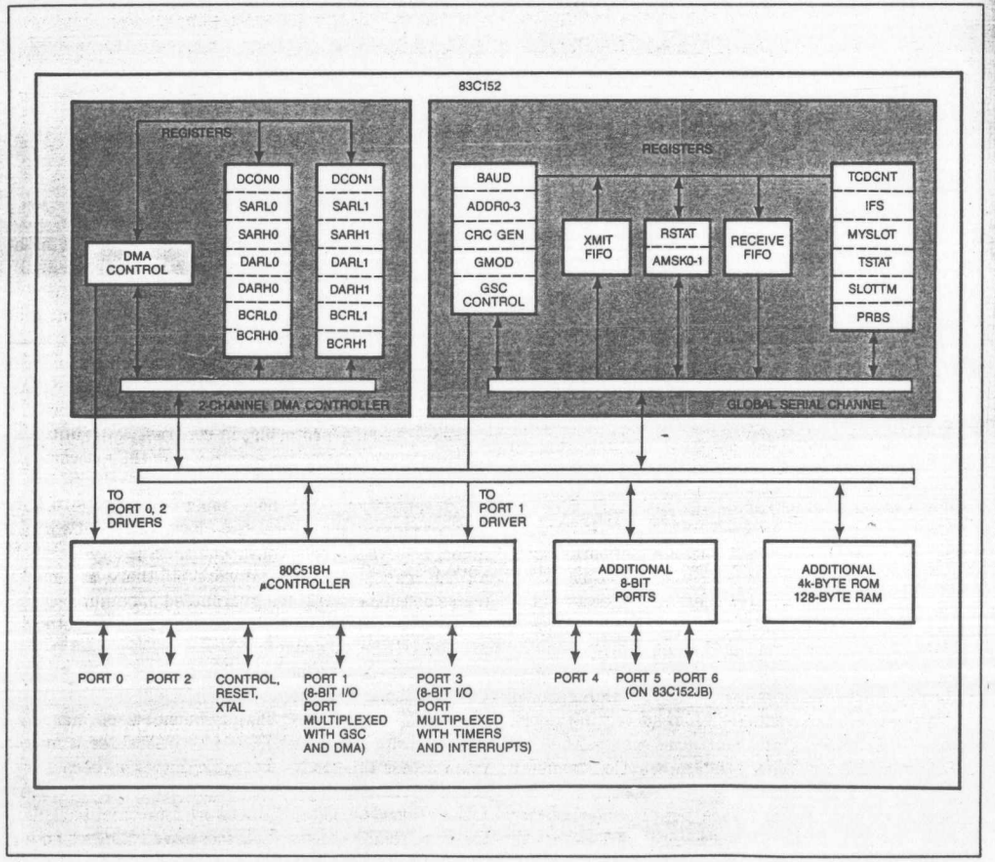

***Fig 1—The 83C152 communication controller** has a multiprotocol Global Serial Channel (GSC) and two DMA channels functioning as peripherals for the 80C51BH µcontroller. Different versions of the 83C152 offer additional ports.*

When a collision takes place, all of the nodes on the network, regardless if they are involved in the collision, enter into something called a resolution phase. The worst-case access time is equal to the maximum time of the resolution phase, which you can calculate from the network's configuration and parameters. During the resolution phase, the protocol assigns each node on the network a programmable time slot for data transmission. During its alloted time slot, a node has the option of transmitting a message. The node's maximum allocated frame length determines the length of the message. This type of recovery sequence eliminates the possibility of successive collisions.

A single IC can do it all

The Intel 83C152 is an example of a single-chip VLSI communication controller that implements the CSMA/DCR protocol (**Fig 1**). (It implements several others as well, including the CSMA/CD protocol.) The IC has an 80C51BH microcontroller core that contains an asynchronous serial port, 4k bytes of ROM, 128 bytes of RAM, four 8-bit parallel-I/O ports, and two 16-bit timer/counters. The IC also has an extra 4k bytes of ROM and an extra 128 bytes of RAM.

The core interfaces to two on-chip peripherals: a 2-channel DMA controller, and a multiprotocol serial channel, the Global Serial Channel or GSC. The DMA channels service both the GSC and the local serial channel. The DMA channels offload the communications overhead from the CPU and arbitrate for shared-memory resources via a hold and a hold-acknowledge bus-access control. The 83C152 interfaces to a serial backplane through open-collector drivers and receivers. To drive larger networks, it uses active drivers such as RS-485 transceivers (**Fig 2**).

Initially, the user code loads the network parameters, including the nodes' slot position, the interframe space (IFS), the slot time (ST), and the maximum number of slots for the network, into the GSC's internal registers. The controller permits 63 nodes on the network to participate in collision resolution. The network can also have an additional node that doesn't participate in collision resolution. The controller must assign slot number 0 to this additional node to prevent it from transmitting during the resolution phase. A data-gathering station would be a typical example of a node with slot number 0.

During the resolution phase, the slot numbers decrement from the maximum number of nodes on the network to slot number 1. The GSC loads the prepro-

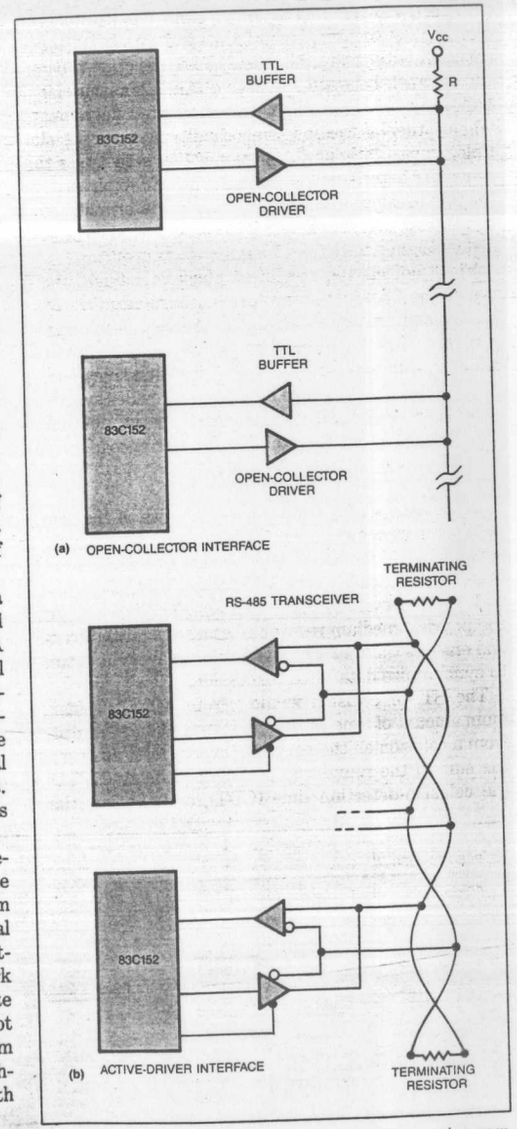

***Fig 2—You can interface** the 83C152 to a backplane using open-collector TTL buffers (**a**). The multiprotocol controller can also interface to larger networks via RS-485 transceivers (**b**).*

During the resolution phase, the CSMA/DCR protocol assigns each node on the network a time slot for data transmission.

grammed maximum number of slots into the decrementing slot counter. It must then wait one IFS after it senses that the link is idle. The IFS is the quiet time on the network between successive frame transmissions. Following this interval, the slot counter decrements when either of the following conditions occur: A slot time expires without detecting any link activity, or the controller detects link activity and an IFS expires after the link becomes idle. When the slot counter matches a node's assigned slot number, the protocol permits the node assigned to that slot number to transmit.

Fig 3 depicts the resolution phase of a 6-node network. The CSMA/DCR protocol encapsulates data into frames similar to the CSMA/CD protocol: Each frame consists of a preamble bit pattern, the destination address, a message field, and cyclic-redundancy check (CRC) bits. The preamble bit pattern provides synchronization data for all the receivers on the network, allowing them to decode the Manchester-encoded data.

Now you can calculate the worst case

The CSMA/DCR protocol relies on two of the same network parameters that the CSMA/CD protocol uses —the IFS and the ST. The network nodes must wait at least one IFS while doing peer-to-peer communications and when recovering from a collision. The IFS also lets the physical medium recover after a transmitted frame and ensures that the other network sublayers have time to synchronize data for transmission.

The ST, or collision window, represents the maximum amount of time that there is invalid data resulting from a collision on the network. It is at least as long as the sum of the round-trip propagation delay (RPTD), the collision-detection time (CDT), and the jam time (JT). The RPTD is equal to twice the network length divided by the propagation speed (approximately 2.1×10^8 msec). The CDT is the time required for the circuitry to detect a collision. The JT lasts for either a 16 or a 32 bit period, depending on the number of CRC bits you program into the 83C152. The 83C152 detects a collision on every bit by verifying whether the transitions of the Manchester code are valid.

TABLE 1—NETWORK PARAMETERS

DATA PROTOCOL	CSMA/DCR
DATA RATE	2M BPS
NETWORK LENGTH	500m
WORST-CASE ACCESS TIME	8 mSEC
INTERFRAME SPACING	96 BIT PERIODS = 48 μSEC
JAM TIME	32 BIT PERIODS = 16 μSEC
COLLISION-DETECTION TIME	1 BIT PERIOD = 0.5 μSEC
SLOT TIME	64 BIT PERIODS = 32 μSEC
ROUND-TRIP PROPAGATION DELAY	4.7 μSEC ≈ 10 BIT PERIODS

The 83C152 programs the IFS and the ST in multiples of the bit period. (A bit period is the inverse of the bit rate). The controller programs each IFS and ST independently; the maximum length of each can be 256 bit periods. During the resolution phase, the calculated worst-case access time is equal to the ST plus an IFS plus the sum of the maximum frame time alloted for each node and an IFS for each frame. In other words:

$$\text{Access Time(WC)}=\text{ST}+\text{IFS}+\Sigma(\text{Maximum Frame Time(i)}+\text{IFS}),$$

where the access time(WC) is the worst-case access time; the ST>RTPD+CDT+JT as stated above; the sum takes into account all the nodes on the network; and the maximum frame time(i) is the maximum frame time of the *i*th node.

When all of the network nodes have the same maximum frame time size, the worst-case access time is

$$\text{Access Time(WC)}=\text{ST}+\text{IFS}+n\times(\text{Maximum Frame Time}+\text{IFS}),$$

where n is the number of nodes on the network. In both formulae, the frame time is equal to the number of bits in a frame divided by the bit rate.

An example will help to illustrate how the network parameters determine the access time. Consider the case of a 2M-bps control network that communicates over a data link having 32 nodes. Two of the nodes have a maximum frame size of 500 bytes, which is typical of a motor controller. The remaining 30 nodes have maximum frame sizes of 20 bytes, which is typical for devices such as sensors and valves. For this sample network, and for the accompanying network parameters listed in **Table 1**, the worst-case access time is approximately 8 msec.

Because the CSMA/DCR protocol guarantees a maximum access time, you can adjust the network parameters for optimum operation. For the same 2M-bps network, **Fig 4** shows the variations in worst-case access time that result when you manipulate two network parameters—the maximum frame size and the number of network nodes. Curves for other network parameters are similar, and therefore you can custom-

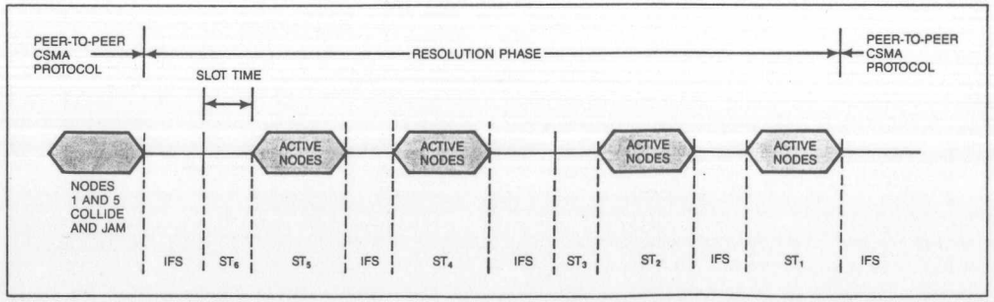

***Fig 3—In this sample configuration of a 6-node network** operating with the CSMA/DCR protocol, nodes 1 and 5 collide during peer-to-peer CSMA communications. When the network becomes idle, all the nodes on the network enter the resolution phase. After an interframe space (IFS), the slot counter decrements from ST_6 to ST_1. Active nodes transmit frames during their respective slot assignments.*

LISTING 1—SAMPLE CODE FOR THE 83C152's GSC

```
MOV GMOD, #7CH          ; select CSMA protocol, Manchester
                        encoding, 32-bit preamble, 32-bit
                        Autodin CRC, alternate back-off,
                        transmitter operation with internal baud
                        generator and 16-bit addressing.
CLR DMA                 ; select GSC operation without DMA
MOV BAUD, #00H          ; set baud rate to 2 Mbps for 16 MHz
                        clock
ANL PCON, #0F7H         ; select receiver operation with
                        internal baud generator
MOV ADR0, #05H          ; set first station address to 1005H
MOV ADR1, #10H
MOV ADR2, #06H          ; set second station address to 2006H
MOV ADR3, #20H
MOV IFS, #060H          ; set Interframe Spacing to 96 bit times
MOV SLOTTM, #40H        ; set Slot Time to 64 bit times
MOV TCDCNT, #20H        ; set maximum number of stations to 32
MOV PRBS, #0FFH         ; freeze the Pseudo-Random Binary
                        Sequence generator for proper DCR
                        operation
MOV MYSLOT, #0C5H       ; select station slot assignment to 5,
                        select DCR mode and DC-type jam.

    Depending on the application, the transmitter and the
receiver can be enabled when required using the following
instructions:

SETB TEN                ; enable the transmitter
SETB GREN               ; enable the receiver
```

ize a network to meet a specified worst-case access time.

At first glance, you might think that you could tailor a network so that the nodes with the higher slot numbers would have lower access times than the nodes with lower slot numbers. After all, the slot counter decrements from a high count to a lower one. Consider the following worst-case scenario, though. The network is in a resolution phase and the slot counter is at a count equal to n. At this time, say, the node with an assigned slot number of n+1 receives an interrupt and wants to

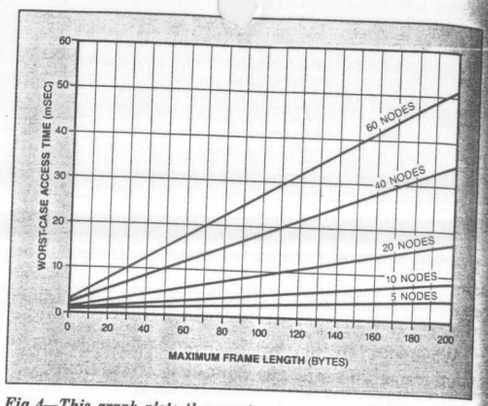

Fig 4—This graph plots the worst-case access time for a 2M-bps CSMA/DCR network having the same length, interframe space, and slot time as the one in Fig 3. As you can see, the worst-case access time varies according to the number of nodes and the maximum alloted frame length per node.

LISTING 2—SAMPLE CODE FOR THE 83C152's DMA CHANNELS

```
SETB DMA                ; select DMA to service transmitter (DMA
                        is a bit in TSTAT register)
MOV DCON0, #88H         ; select DMA channel 0 for transmitter,
                        DMA demand mode by serial channel and
                        external memory as source.  The GO bit
                        is not set yet.
MOV DCON1, #29H         ; select DMA channel 1 for receiver, DMA
                        demand mode by serial channel and
                        external memory as destination.  The GO
                        bit is set.
MOV SARH0, #10H         ; Source for channel 0 is memory
                        starting at
MOV SARL0, #00H         ;    address 1000H
MOV DARH0, #00H         ; Destination for channel 0 is Transmit
                        FIFO
MOV DARL0, #85H
MOV BCRH0, #00H         ; Transmit a count of 10 bytes
MOV BCRL0, #0AH
MOV SARH1, #00H         ; Source for channel 1 is Receive FIFO
MOV SARL1, #0F4H
MOV DARH1, #20H         ; Destination for channel 1 is memory
MOV DARL1, #00H         ;    starting at address 2000H
MOV BCRH1, #00H         ; receive up to 64 bytes
MOV BCRL1, #40H
```

Depending on the application, the transmitter and the receiver are enabled when required using the following instructions:

```
SETB TEN                ; enable the transmitter and then set
                        the GO
ORL DCON0, #01H         ;      bit for the DMA to start
                        transmission.
SETB GREN               ; enable the receiver
```

transmit immediately. Before the node can transmit, the current resolution phase must count out, and if by chance another collision occurs at the end of that phase, a new resolution phase must begin to decrement. The access time for the n+1 node would then be the worst-case calculated value.

Once you've determined your network's configuration, the 83C152 lets you establish network parameters easily. **Listing 1** shows the code that initializes the GSC to perform the DCR operation for a particular node. Programmable features of the GSC include the bit rate (from 2M bps with an internal baud generator to 2.5M bps with an external clock), the preamble bit pattern, multiple destination addresses, and the type of CRC bits. **Listing 2**'s program shows how you can use the two DMA channels to service the GSC. **EDN**

Author's biography

Deif N Atallah is a staff engineer with Intel's Embedded Operation Group in Chandler, AZ. He is currently responsible for the product architecture of 32-bit embedded controllers. He obtained his BSEE from Alexandria University, Egypt, and his MSEE degree from Penn State University. Deif enjoys world traveling, camping, and photography.

Article Interest Quotient (Circle One)
High 497 Medium 498 Low 499